AFRL-IF-RS-TR-2002-65
Final Technical Report
April 2002

SUPPORT FOR ADVANCED SOFTWARE ENGINEERING ENVIRONMENT

University of Massachusetts

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. B127

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2002-65 has been reviewed and is approved for publication.

APPROVED: Roger J Dziegiel, Jr

ROGER J. DZIEGIEL
Project Engineer

FOR THE DIRECTOR: Michael Talbert

MICHAEL TALBERT, Maj., USAF, Technical Advisor
Information Technology Division
Information Directorate

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. AGENCY USE ONLY (Leave blank)	**2. REPORT DATE** APRIL 2002	**3. REPORT TYPE AND DATES COVERED** Final Jun 94 – Jun 99

4. TITLE AND SUBTITLE SUPPORT FOR ADVANCED SOFTWARE ENGINEERING ENVIRONMENT	**5. FUNDING NUMBERS** C - F30602-94-C-0137 PE - 62301E PR -: B127 TA - 01 WU - 01
6. AUTHOR(S) Leon J. Osterweil and Lori A. Clarke	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Massachusetts Department of Computer Science 140 Governor's Drive Amherst Massachusetts 01003-4610	**8. PERFORMING ORGANIZATION REPORT NUMBER** N/A
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Advanced Research Projects Agency AFRL/IFTD 3701 North Fairfax Drive 525 Brooks Road Arlington Virginia 22203-1714 Rome New York 13441-4505	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER** AFRL-IF-RS-TR-2002-65

11. SUPPLEMENTARY NOTES
AFRL Project Engineer: Roger Dziegiel/IFTD/(315) 330-2185

12a. DISTRIBUTION / AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.	**12b. DISTRIBUTION CODE**

13. ABSTRACT *(Maximum 200 Words)*

The goal of software analysis research is to develop practical techniques that can help software developers determine whether software systems satisfy their requirements. The focus of most of the research is on the static analysis of concurrent software—the nondeterministic behavior introduced by concurrency means that dynamic analysis (testing) is not adequate for concurrent systems. Since concurrent systems are built of interacting sequential components, many of the techniques used for analyzing concurrent systems can also be applied to sequential software.

FLAVERS is an example of a flexible, powerful system for automatically guaranteeing the absence or detecting the presence of a wide range of user-specified properties or behaviors in both sequential and concurrent systems. FLAVERS complements traditional testing approaches, which only demonstrate the presence or absence of errors for the specific test cases that have been executed. It also complements formal verification methods, which employ more comprehensive analysis, but require extensive expertise on part of the user.

14. SUBJECT TERMS Software Architectures, High Assurance Software, Software Process	**15. NUMBER OF PAGES** 7
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	**18. SECURITY CLASSIFICATION OF THIS PAGE** UNCLASSIFIED	**19. SECURITY CLASSIFICATION OF ABSTRACT** UNCLASSIFIED	**20. LIMITATION OF ABSTRACT** UL

TABLE OF CONTENTS

Summary

The Arcadia Project involved researchers at the University of Massachusetts, Amherst, the University of California at Irvine, and the University of Oregon. The products of the Arcadia Project are categorized in three basic areas: analysis and testing technology and tools; process technology and artifacts; and technology to support the deployment of perpetual analysis and testing tools and processes. The primary foci of the University of Massachusetts effort was on the improvement of static analysis technology and tools, on integration of static and dynamic analysis techniques, and the development of process technologies and artifacts. In this final report the major accomplishments are described and some of the technology transition efforts are listed.

1.0 The FLAVERS finite state verification system:
FLAVERS uses data flow analysis techniques to verify user-specified properties. It is not as general as theorem proving based verification techniques, but it is more efficient, always terminates, and requires considerably less human expertise. During this contract, the FLAVERS prototype was significantly improved. The major accomplishments included:

- Optimizing the FLAVERS analysis engine, improving performance by nearly a factor of 100. This greatly increased the size and complexity of the programs that can be verified.

- Extending FLAVERS to handle the concurrency control constructs used in Java. Completed a preliminary evaluation that demonstrated the ability to find the several error-prone concurrency control patterns that can easily arise using Java.

- Developing algorithms for both Java and for Ada that can efficiently compute the statements from different tasks that might execute concurrently. Completed an experimental evaluation that showed that our algorithms are more efficient and precise than previously known algorithms. This information is used to improve the model of the program that FLAVERS generates but it is also useful for compiler optimizations.

- Investigated alternative reasoning algorithms optimized for different phases in the software testing and analysis process.

- Completed a finite state automata (FSA) toolset that supports the specification of properties and constraints directly as FSAs as well as the translation of specifications written in qualified regular expressions. Properties are checked for well-formedness, can be visualized, and can be executed.

- Provided capabilities for visualizing counter example paths created by FLAVERS to show where the property violation occurs.

2.0 Little-JIL Process Programming Language:
Little-JIL is a high-level, visual agent coordination language that succinctly represents complex computer and human agent interaction. It provides support for a range of exception handling capabilities and resource management. During the course of this contract, the Little-JIL language was refined and an interpreter, called Juliet, was developed. The Juliet interpreter is being developed as a distributed system of servers that support such key functions as software artifact management, consistency management, resource management, and scheduling. Major Little-JIL/Juliet accomplishments included:

- Developing a visual editor for the language.

- Designing a general resource specification, allocation, and management system for human and tool requests.

- Developing an agenda management system that maintains and coordinates the worklist for multiple, distributed agents. This system is a general coordination system for distributed system and is used in Juliet to support the communication and cooperation among the agents participating in a process.

- Developed an approach for integrating a process program for a complex toolset with the GUIs provided by that toolset. Using this approach, process programmers can specify how processes are to react to GUI events and vice versa. This specification is then used to generate a mediator that will enforce this interaction. This approach and the generator were demonstrated using the FLAVERS toolset.

- Several process programs were developed to evaluate and demonstrate the approach. Some of these include: a perpetual bug-tracking process, a regression testing process, a process to support FLAVERS, an object –oriented design process.

3.0 Technology transition efforts included:

- Northrop B-2 Division, using the UMASS language independent language processing toolset, provided support for Jovial programs and extended FLAVERS to verify Jovial B-2 software.

- The MCC Quest project, using the UMASS language independent language processing toolset, provided support for C++ programs and extended FLAVERS to verify C++ software. This version of the system was distributed to the MCC Quest project participants.

- Honeywell applied FLAVERS and perpetual testing processes to avionics software.

- TACOM, TARDEC, adopted a perpetual test program based on our Perpetual Testing project.

- SAIC used FLAVERS to demonstrate the presence and absence of errors in Stricom distributed simulation code.

- Lockheed Martin used JIL to model the DAGAR process, demonstrating the benefits of JIL for capturing realistic industrial processes.

- TASC developed a demonstration V&V systems for avionics software using the UMASS language independent language processing toolset and the UCI ProDAG system